

FIG. 1

FIG. 2A

```
1 <!-- BEGIN Event Tracking Code -->
2 <script language="javascript1.2">
3 function recordit(k, w)
4 //This function records the link. Parameters k and w are passed. K is the link and w is an array of stored
5 functions.
6 {
7   var gettext="NA";
8   //This initializes the variable used to hold the text that the link wrapped.
9   var iloc=0;
10  while (document.links[iloc]!==k) {
11    iloc++;
12  }
13  //This is a loop which determines which element in the array the link that was passed.
14  if (w[iloc]){w[iloc]();}
15  //This checks to see if a function was stored for that link. If so, it executes the stored function.
16  document.all?gettext=k.innerHTMLHTML:gettext=k.text;
17  //This sets the gettext variable to the text or html that was wrapped by the link.
18  document.images.hitboximage.src="http://ww3.hitbox.com/logs/log.cgi?link="+k.href+"&iloc="+iloc+"
19  &named="+gettext;
20  //This sends the link information back to the hitbox servers.
21  }
22
```

FIG. 2B

```

23 function initial()
24 //This function sets the onclick event for each link and stores prior events associated with it, if any.
25 {
26     var storef = Array(document.links.length);
27     //This array will store prior functions.
28     for (var i = 0; i < document.links.length; i++) {
29         //This loop runs through all the documents on a page.
30         if (document.links[i].onclick) storef[i]=document.links[i].onclick;
31         //If a function is associated with that link it is stored in an array.
32         document.links[i].onclick = function () {
33             recordit(this, storef);
34             return true;
35             //Here is where the onclick event for each link is associated with the recordit function.
36         }
37     }
38 }
39
40
41 window.onload = initial;
42 //This starts the process when the window has loaded.
43 </script>
44
45 <IMG SRC="http://stats.hitbox.com/buttons/ct0.gif" HEIGHT=68 WIDTH=82 BORDER=0
46 NAME="hitboximage">
47 <!-- This is the initial image used to record a page view.

```

FIG. 3

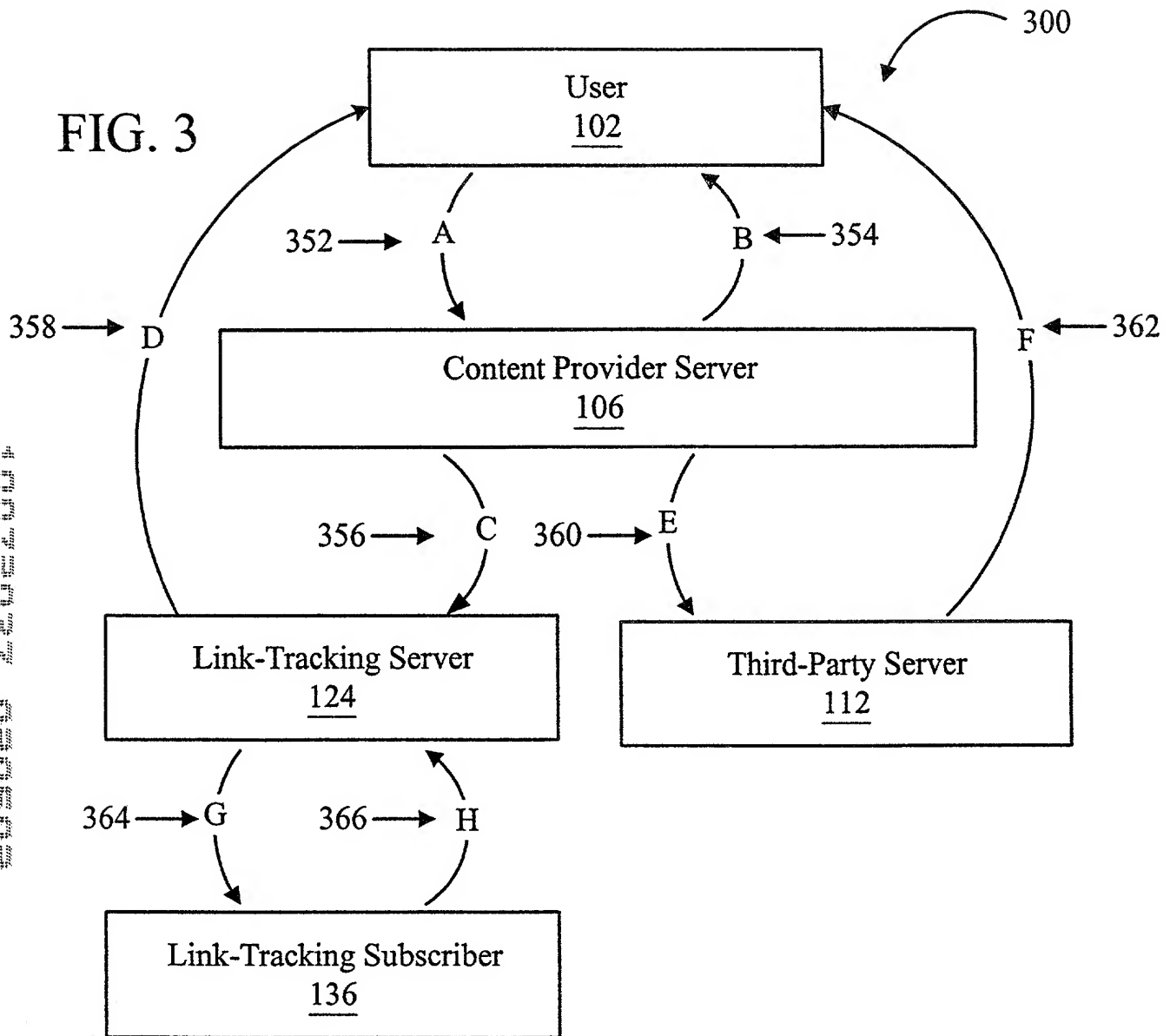


FIG. 4A

```
1 <script language="javascript1.2">
```

```
2 This code demonstrates how to record downloads and exit links using the invention.
```

```
3 _hbi=new Image();
```

```
4 _ot=0;
```

```
5 //Initializes variables.
```

```
6
```

```
7 function _gf(_x){
```

```
8 //This function retrieves the file or document name requested in a URL.
```

```
9     _ed = (_x.length);
```

```
10    _be = ((_x.lastIndexOf("/"))+1);
```

```
11    _fn = (_x.substring(_be, _ed));
```

```
12    return _fn;
```

```
13 }
```

```
14
```

```
15 function _sb(_k){
```

```
16 //This function checks to see if the link is an exit link.
```

```
17    eval("var _gt=(/\\w*\\.)*(\\w*\\.)/i;");
```

```
18    _gt.exec(location.hostname);
```

```
19    //This parses the link for domain name.
```

```
20    var _fr="co.com.org.net.";
```

```
21    if(_fr.indexOf(RegExp.$2)!=-1){
```

```
22        eval("var _uf=(/\\w*\\.)*(\\w*\\.)(\\w*\\.)/");
```

```
23        _uf.exec(location.hostname);}
```

```
24    //This is additional checking in case the link is to a foreign domain.
```

```
25    if(_k.href.indexOf(RegExp.$2)==-1){_hbi.src="http://ehg.hitbox.com/HG?el="+escape(_k.href);}
```

```
26    //If the domain name is not the same as the current document, the information is sent back to the hitbox servers.
```

```
27    return true;
```

```
28 }
```

```
29
```

FIG. 4B

```

30 function _ri(_k,_w){
31 //This function receives the link and checks if it is a download. Parameters _k and _w are passed. _k is the
32 link and _w is an array of stored functions.
33
34 var _ilc=0;
35 while(document.links[_ilc]!=$_k){ _ilc++;}
36 if(_w[_ilc]){ _w[_ilc]();}
37 //This is a loop which determines which element in the array the link that was passed.
38 _fl=_gf(_k,pathname);
39 //Sets variable _fl to the file or document name in the link.
40 if(_fl!=""){
41 //Checks if name exists
42 var _pt=new RegExp("\\.?html?\\.asp|\\.cfm|\\.jsp|\\.cgi|\\.php[3-4]?\\.p|\\.taf\\.dll","i");
43 if(!_pt.test(_fl)){
44 //Checks to see if file extension is among a list of known HTML-types.
45 _hbi.src="http://ehg.hitbox.com/HG?fn="+escape(_fl);}
46 //If the file is not among that list, the file information is sent back to the hitbox servers.
47 else{
48     _sb(_k);}}
49 //If info was not sent back to hitbox server, it checks for exit links.
50 else{
51     _sb(_k);
52 //If info was not sent back to hitbox server, it checks for exit links.
53 }
54 return true;
55 }

```

FIG. 4C

```

56 function _it(){
57     //This function sets the onclick event for each link and stores prior events associated with it, if any.
58     if(_ot==1)_wo();
59     //If an onload event was previously set, executes it now.
60     var _sf=new Array(document.links.length);
61     //This array will store prior functions.
62     for(var i=0;i<document.links.length;i++){
63         _rv="true";
64         if(document.links[i].onclick){
65             //This loop runs through all the documents on a page.
66             _sf[i]=document.links[i].onclick;
67             //If a function is associated with that link it is stored in an array.
68             if(_sf[i].toString().indexOf("return false")!=-1)_rv="false";
69             //If the return value of the stored function is false, the return value for the new function is
70             set to false.
71         }
72     }
73     eval("document.links[i].onclick=function(){_ri(this,_sf);return "+_rv+";}");
74     //Here is where the onclick event for each link is associated with the _ri function.
75 }
76
77 if(window.onload){_wo=window.onload;_ot=1;};window.onload=_it;
78 //If a function is already associated with the window loading, it is stored and executed later.
79 </script>

```